KDI
RDF

Fausto Giunchiglia and Mattia Fumagallli

University of Trento

Roadmap

- XML
  - From HTML to XML
  - Similarities and differences
  - XML features and limits

- RDF
  - Syntax
    - Language and data model
    - Reification
    - Containers and collections
  - Semantics
    - RDF schema
  - Reasoning
    - Inference system
    - Inference rules

  - Exercises

XML

HTML vs. XML

The Adventures of Tom Sawyer

Front piece of *The Adventures of Tom Sawyer*

Author	Mark Twain
Cover artist	created by Mark Twain
Country	United States
Language	English, Limited Edition(Spanish)
Genre	Bildungsroman, picaresque, satire, folk, children's novel
Publisher	American Publishing Company
Publication date	1876[1]
OCLC	47052486
Dewey Decimal	Fic. 22
LC Class	PZ7.T88 Ad 2001
Followed by	*Adventures of Huckleberry Finn*
Text	*The Adventures of Tom Sawyer* at Wikisource

HTML: focus on presentation

```
<h2>The adventures of Tom Sawyer</h2>
…
<b>Author: </b> Mark Twain <br>
<b>Cover artist: </b> created by <a href="http://…">Mark Twain
</a>
…
```

XML: focus on metadata

```
<book>
        <title> The adventures of Tom Sawyer </title>
        <author> Mark Twain </author>
        <genre> Bildungsroman </genre>
        <genre> picaresque </genre>
        …
        <publisher> American Publishing Company </publisher>
        <year>1876</year>
</book>
```

HTML vs. XML: similarities and differences

Similarities

- They both use of tags
- Tags may be nested
- Human can read and interpret both HTML and XML quite easily
- Machines can read and interpret only to some extent

Differences

- HTML is to tell machines about how to interpret formatting for graphical presentation
- XML is to tell machines about metadata content and relationships between different pieces of information
- XML allows the definition of constraints on values
- HTML tags are fixed, while XML tags are user defined

More about XML

- **XML meta markup language: language for defining markup languages**

- **Query languages for XML:**
  - **Xquery**
  - **XQL**
  - **XML-QLs**

- **Style sheets can be written in various languages to define how to present XML to humans:**
  - **CSS2 (cascading style sheets level 2)**
  - **XSL (extensible stylesheet language)**

- **Transformations: XSLT specifies rules to transform an XML document to:**
  - **another XML document**
  - **an HTML document**
  - **plain text**

XML in a nutshell

XML features:

- A metalanguage that allows users to define markup
- It separates content and structure from formatting
- It is the de facto standard for the representation and exchange of structured information on the Web
- It is supported by query languages

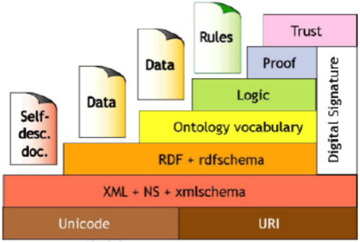
XML limits:

- The semantics of XML documents is not accessible to machines
- The nesting of tags does not have standard meaning
- Interoperability is possible if there is a shared understanding of the vocabulary

RDF

RDF and the Semantic Web

RDF (Resource Description Framework) is at the basis of the Semantic Web



Definitions

- A language for representing Web resources and information about them in the form of metadata [RDF Primer]
- A language to represent all kinds of things that can be identified on the Web [RDF Primer]
- A domain independent data model for representing information on the Web [G. Antoniou and F. van Harmelen, 2004]
- A language with an underlying model designed to publish data on the Semantic Web [F. Giunchiglia et al., 2010]

Distributing Data Across the Web

WORKS				
ID	**Title**	**Author**	**Medium**	**Year**
1	Hamlet	Shakespeare	Play	1599
2	Othello	Shakespeare	Play	1604
3	Edward II	C. Marlowe	Play	1592
4	Hero and Leander	C. Marlowe	Poem	1593

SUBJECTS

VALUE

PROPERTIES

Data Distribution (over many machines where each machine maintains a part)

- **row by row**
- **column by column**
- **cell by cell** (the strategy taken by RDF):
  - a global identifier for the column headings
  - a global identifier for the row headings
  - a global identifier for non-literal values

RDF syntax

RDF language and data model

RDF language

- A language for representing data in the Semantic Web
- A language for expressing simple statements of the form subject-property-value (binary predicates)
- The capability to perform inference on statements

RDF data model

- The data model in RDF is a graph data model
- An edge with two connecting nodes forms a triple

RDF language

Formal syntax:

- RDF has been given a syntax in XML and inherits all its benefits
- Statements in RDF are machine comprehensible

Resources:

- An object, an entity or anything we want to talk about (e.g. authors, books, publishers, places, people, facilities)

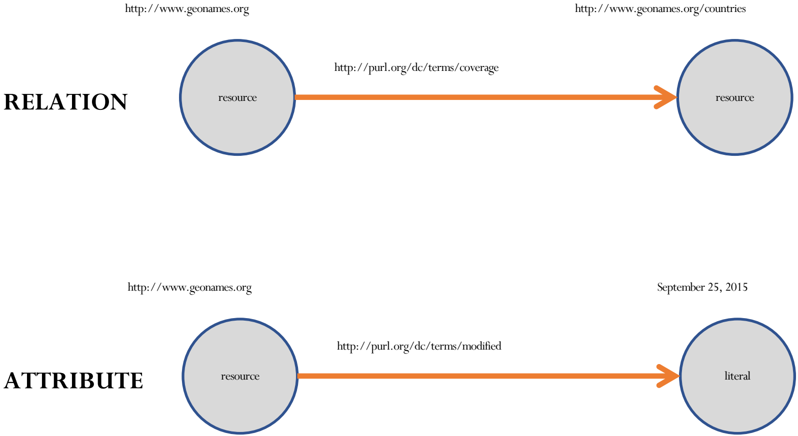
Properties:

- They codify relations (e.g. written-by, friend-of, located-in, …) and attributes (e.g. age, date of birth, length …)

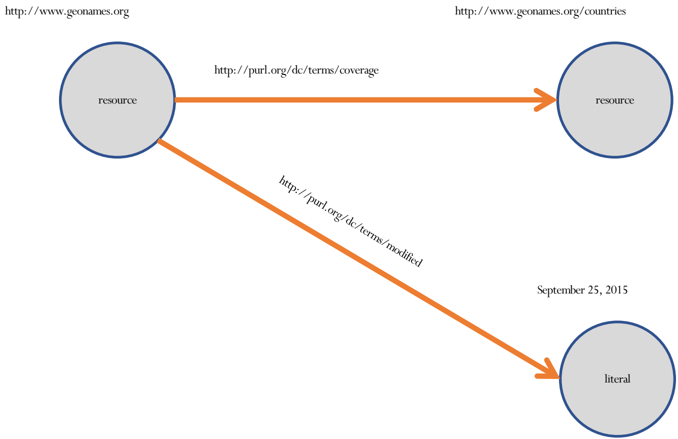
Statements:

- Statements assert the properties of resources in form of triples subject-property-value
- Every resource and property has a URI (an URL or any other identifier)
- Values can be resources (for relations) or literals (for attributes)

RDF data model



RDF as graph



http://www.geonames.org
http://www.geonames.org/countries
http://purl.org/dc/terms/coverage
resource
resource
http://purl.org/dc/terms/modified
September 25, 2015
literal

XML syntax example

```
<?xml version="1.0"?>

 <rdf:RDF
                        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
                        xmlns:dc="http://purl.org/dc/terms#">

<rdf:Description rdf:about="http://www.geonames.org">
     <rdfs:label>GeoNames</rdfs:label>
     <dc:coverage rdf:resource="http://www.geonames.org/countries"/>
     <dc:modified>September 25, 2015</dc:modified>
</rdf:Description>

</rdf:RDF>
```

RDF/XML elements

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:uni="http://www.mydomain.org/uni-ns">

    <rdf:Description rdf:about="CIT1111">
        <uni:courseName>Discrete Matematics</uni:courseName>
        <uni:isTaughtBy rdf:resource="#949318">
        <uni:age rdf:datatype="&xsd:integer">27<uni:age>
    </rdf:Description>

    <rdf:Description rdf:ID="#949318">
        <uni:name>David Billington</uni:name>
        <uni:title>Associate Professor</uni:title>
        <uni:age rdf:datatype="&xsd:integer">27<uni:age>
    </rdf:Description>
</rdf:RDF>
```

NAMESPACES

URI or fragment of it

VALUE

RESOURCE HAS BEEN DEFINED ELSEWHERE

RELATION

ATTRIBUTE

DATA TYPE

RESOURCE IS DEFINED HERE

RDF typing

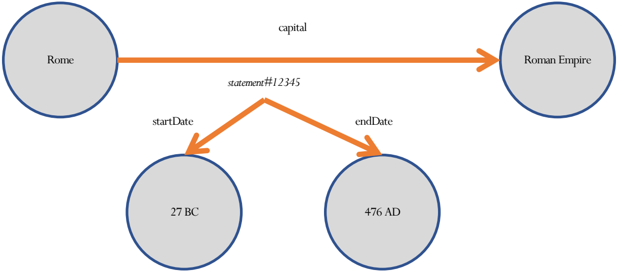
```
<rdf:Description rdf:ID="CIT1111">
        <rdf:type rdf:resource="http://www.mydomain.org/uni-
    ns#course"/>
        <uni:courseName>Discrete Maths</uni:courseName>
        <uni:isTaughtBy rdf:resource="#949318"/>
</rdf:Description>

<rdf:Description rdf:ID="949318">
        <rdf:type rdf:resource="http://www.mydomain.org/uni-
    ns#lecturer"/>
        <uni:name>David Billington</uni:name>
        <uni:title>Associate Professor</uni:title>
</rdf:Description>
```

RDF Reification

Reification can be used to represent:

- Generic statements about statements
- Structured attributes (e.g. address)
- Units of measure
- Provenance information
- Time validity and other contextual information



RDF Reification: example

In the following it represents the fact that "the item 10245 (basically a tent) has weight 2.4 (in some measuring unit, e.g., kg)" and "the staff with id 85740 has written this statement"

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:exterms="http://www.example.com/terms/"
        xml:base="http://www.example.com/2002/04/products">

 <rdf:Description rdf:ID="item10245">
   <exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
 </rdf:Description>

 <rdf:Statement rdf:about="#triple12345">
   <rdf:subject           rdf:resource="http://www.example.com/2002/04/products#item10245"/>
   <rdf:predicate rdf:resource="http://www.example.com/terms/weight"/>
   <rdf:object rdf:datatype="&xsd;decimal">2.4</rdf:object>

   <dc:creator rdf:resource="http://www.example.com/staffid/85740"/>
 </rdf:Statement>

</rdf:RDF>
```

Below the URI **triple12345** is used to identify the original statement provided above

Containers

Containers collect a number of resources or attributes about which we want to make statements as a whole:

- **rdf:Bag** an unordered container (e.g. members of a group, documents in a folder)
- **rdf:Seq** an ordered container (e.g. modules of a course, items on an agenda, an alphabetized list of staff members)
- **rdf:Alt** a set of alternatives (e.g. the document home and mirrors, translations of a document in various languages)

Members are listed using rdf:li or by rdf:_1, rdf:_2 …

Containers are open lists of members, i.e. we cannot exclude the existence of other members

Examples (in compact representation)

```
<uni:lecturer rdf:ID="949352" uni:name="Grigoris Antoniou"
        uni:title="Professor">
        <uni:coursesTaught>
                <rdf:Bag rdf:ID="DBcourses">
                        <rdf:_1 rdf:resource="#CIT1112"/>
                        <rdf:_2 rdf:resource="#CIT3116"/>
                </rdf:Bag>
        </uni:coursesTaught>
</uni:lecturer>
```

```
<uni:course rdf:ID="CIT1111"        uni:courseName="Discrete Mathematics">
        <uni:lecturer>
                <rdf:Alt>
                        <rdf:li rdf:resource="#949352"/>
                        <rdf:li rdf:resource="#949318"/>
                </rdf:Alt>
        </uni:lecturer>
</uni:course>
```

Collections

Collections can represent a close list of members overcoming the limitation of Containers

Example: a collection with exactly 3 members:

```
<rdf:Description rdf:about="#CIT2112">
    <uni:isTaughtBy rdf:parseType="Collection">
        <rdf:Description rdf:about="#949111"/>
        <rdf:Description rdf:about="#949352"/>
        <rdf:Description rdf:about="#949318"/>
    </uni:isTaughtBy>
</rdf:Description>
```

RDF semantics

RDF schema

RDF

- **RDF is a universal language** that lets users describe resources in their own vocabularies
- RDF by default does not assume, nor defines semantics of any particular application domain

RDF schema (RDFS)

A language defined to provide mechanisms to add semantics to RDF resources, in terms of:

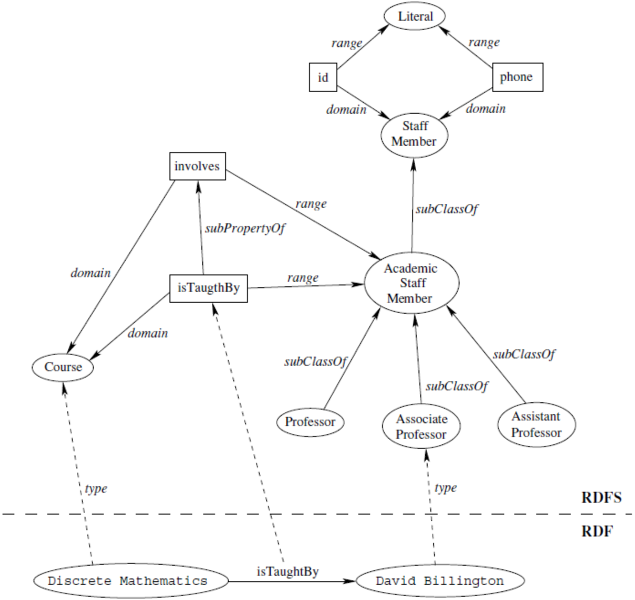
- Classes (*rdfs:Class*) and Properties (*rdfs:Property*)
- Class Hierarchies and Inheritance (*rdfs:subClassOf*)
- Property Hierarchies (*rdfs:subPropertyOf*)
- Domain (*rdfs:domain*) and range (*rdfs:range*) of properties

It is similar to the object-oriented programming (OOP) paradigm with the difference that in OOP the central notion is the class (and properties are defined for them), while in RDF the central notion is the property and classes are used to specify their domain/range.

Classes and instances

Individual objects that belong to a class are referred to as **instances** of that class (*rdf:type*).

Graphical example



RDF example

```
<rdfs:Class rdf:about="#lecturer">
        <rdfs:subClassOf rdf:resource="#staffMember"/>
</rdfs:Class>
<rdf:Property rdf:ID="phone">
        <rdfs:domain rdf:resource="#staffMember"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
```

Utility properties

- **rdfs:seeAlso** relates a resource to another resource that explains it
- **rdfs:isDefinedBy** is a subproperty of rdfs:seeAlso and relates a resource to the place where its definition, typically an RDF schema, is found
- **rdfs:comment** support comments that can be associated with a resource
- **rdfs:label** is a human-friendly name associated with a resource

```
<rdfs:Class rdf:ID="course">
        <rdfs:comment>The class of courses</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="isTaughtBy">
        <rdfs:comment>
                Inherits its domain ("course") and range ("lecturer")
                from its superproperty "involves"
        </rdfs:comment>
        <rdfs:subPropertyOf rdf:resource="#involves"/>
</rdf:Property>
```

Reasoning

Inference system

Sound and complete set of inference rules:

- The RDF inference system consists of **inference rules**
- **Sound**: inference rules prove only formulas that are valid with respect to its semantics
- **Complete**: every formula having a certain property can be derived) inference systems

Examples of rules:

(transitivity)
IF E contains the triples (?u,rdfs:subClassOf,?v) and (?v,rdfs:subclassOf,?w)
THEN E also contains the triple (?u,rdfs:subClassOf,?w)

(inheritance)
IF E contains the triples (?x,rdf:type,?u) and (?u,rdfs:subClassOf,?v)
THEN E also contains the triple (?x,rdf:type,?v)

RDF Inferencing by example

Type (rdf:type) propagation through rdfs:subClassOf

:Fausto Giunchiglia	rdf:type	:Professor
:Professor	rdfs:subClassOf	:Faculty
:Fausto Giunchiglia	rdf:type	:Faculty (inferred)

Relationship propagation through rdfs:subPropertOf

:professorshipAt	rdfs:subProperytOf	:affiliationWith
:Fausto Giunchiglia	:professorshipAt	:UniTN
:Fausto Giunchiglia	:affiliationWith	:UniTN (inferred)

Type identification through rdfs:domain

:professorshipAt	rdfs:domain	:Person
:Fausto Giunchiglia	:professrshipAt	:UniTn
:Fausto Giunchiglia	rdf:type	:Person (inferred)

RDF Inferencing by example

Type identification through rdfs:range

:professorshipAt	rdfs:range	:Educational_Institution
:Fausto_Giunchiglia	:professrshipAt	:UniTn
:UniTn	rdf:type	:Educational_Institution (inferred)

Inferencing through rdfs:domain and rdfs:subClassOf

:Researcher	rdfs:subClassOf	:Scientist
:hIndex	rdfs:domain	:Researcher
:Fausto Giunchiglia	:hIndex	44
:Fausto Giunchiglia	rdf:type	:Researcher (inferred)
:Fausto Giunchiglia	rdf:type	:Scientist (inferred)

Inferencing through rdfs:range and rdfs:subClassOf

:Educational_Institution	rdfs:subClassOf	:Organization
:professorshipAt	rdfs:range	: Educational Institution
:Fausto Giunchiglia	:professorshipAt	:UniTn
:UniTn	rdf:type	:Educational Institution (inferred)
:UniTn	rdf:type	:Organization (inferred)

Intersection and union in RDF

Set Intersection (if an entity e is in X, it is also in both Y and Z)

X	rdfs:subClassOf	Y
X	rdfs:subClassOf	Z
e	rdf:type	X
e	rdf:type	Y (inferred)
e	rdf:type	Z (inferred)

Set Union (any entity e that belongs either to Y or Z also belongs to X)

Y	rdfs:subClassOf	X	
Z	rdfs:subClassOf	X	
e	rdf:type	Y	or
e	rdf:type	Z	
e	rdf:type	X (inferred)	

Summary

Summary

- RDF provides a foundation for representing and processing metadata
- RDF has a graph-based data model
- RDF has a (XML-based) syntax and a semantics (via RDF Schema)
- RDF has a decentralized philosophy and allows incremental building of knowledge, and its sharing and reuse across the Web

- RDF is domain-independent
- RDF Schema provides a mechanism for describing specific domains
- RDF Schema is a primitive ontology language
- It offers certain modelling primitives with fixed meaning
- There exist query languages for RDF and RDFS, including SPARQL

Exercises

Exercise 1

Produce an RDF triple representation of the product, manufacturer and stock information provided in the following table.

ID	Model Number	Division	Product Line	Manufacturing Location	SKU	Available
1	RT-11	Safety	Safety valve	Trento	LM5647	70
2	RTX-56	Safety	Safety valve	Trento	DK3852	30
3	MBB-32	Accessories	Monitor	Hong Kong	CM7823	50
4	DR-43	Control Engineering	Sensor	Malaysia	SN2643	30

Table: Product

Solution

Subject	Predicate	Object
product:Product1	product:id	1
product:Product1	product:modelNumber	RT-11
product:Product1	product:division	Safety
product:Product1	product:productLine	Safety Valve
product:Product1	product:manufacturingLocation	Trento
product:Product1	product:sku	LM5647
product:Product1	product:available	70
product:Product2	product:id	2
product:Product2	product:modelNumber	RTX-56
…		

Exercise 2

Applications that use RDF data from multiple sources need to overcome the issue of managing terminology.

Suppose that one source uses the term *analyst* and another one uses the term *researcher*. How can you represent the fact that:

2.1) *researcher* is a special case of *analyst*?

2.2) *researcher* and *analyst* may overlap?

2.3) *researcher* and *analyst* are equivalent?

Solution

2.1) If a researcher is a special case of analyst, then all researchers are also analysts. This kind of "if/then" relationship can be represented with a single rdfs:subClassOf relation.

:Researcher	**rdfs:subClassOf**	**:Analyst**

2.2) In this case we can define a new class and express the fact that both classes specialize it (so they may overlap).

:Researcher	**rdfs:subClassOf**	**:Investigator**
:Analyst	**rdfs:subClassOf**	**:Investigator**

2.3) RDFS does not provide a primitive construct for expressing class equivalence. However, it can be represented using rdfs:subClassOf.

:Analyst	**rdfs:subClassOf**	**:Researcher**
:Researcher	**rdfs:subClassOf**	**:Analyst**

Exercise 3

Model the following problem in RDF:

"A military mission planner wants to determine off-limits areas, i.e. areas that cannot be targeted by weapons. There are two sources of information contributing to the decision. One source says that civilian facilities (e.g. churches, schools and hospitals) must never be targeted. Another source provides information about off-limits airspaces, called no-fly zones."

Solution

source1:CivilianFacility	**rdfs:subClassOf**	**mmp:OffLimits**
source2:NoFlyZone	**rdfs:subClassOf**	**mmp:OffLimits**

Exercise 4

Suppose an application imports RDF data from an excel file.

- There are two classes of entities, Person and Movie, defined by the import.
- For Person a property called *personName* is defined that gives the name by which that person is known.
- For Movie, the property called *movieTitle* gives the title under which the movie was released.

How to use the standard property *rdfs:label* to develop a generic display mechanism for showing both the names of the persons and titles of the movies?

Solution

We can define each of the properties as subproperty of rdfs:label

personName	**rdfs:subPropertyOf**	**rdfs:label**
movieTitle	**rdfs:subPropertyOf**	**rdfs:label**

Exercise 5a

Consider that a shipping company has a fleet of vessels including:

- new ones that are under construction
- old ones that are being repaired
- the ones that are currently in service
- the ones that have been retired from service

Represent information in the table in RDF

Name	Maiden Voyage	Next Departure	Decommission Date	Destruction Date
Titanic	April 10, 1912			April 14, 1912
MV 16	May 23, 2001	November 29, 2013		
MV 22	June 8, 1970		February 10, 1998	

Table: Ships

Solution

RDF statements to be produced include:

ship:Titanic	ship:destructionDate	“April 14, 1912”
ship:MV16	ship:nextDeparture	“November 29, 2013”
ship:MV22	ship:maidenVoyage	“June 8, 1970”

The following statements hold between classes:

ship:DeployedVessel	rdfs:subClassOf	ship:Vessel
ship:InServiceVessel	rdfs:subClassOf	ship:Vessel
ship:OutOfServiceVessel	rdfs:subClassOf	ship:Vessel

Exercise 5b

How can we represent the following inferences?:

- **if a vessel has a maiden voyage, then it is a Deployed Vessel**
- **if next departure date is set, then it is a In Service Vessel**
- **if it has decommission date or destruction date, then it is a Out Of Service Vessel**

Solution

ship:maidenVoyage	rdfs:domain	ship:DeployedVessel
ship:nextDeparture	rdfs:domain	ship:InServiceVessel
ship:decommisionDate	rdfs:domain	ship:OutOfServiceVessel
ship:destructionDate	rdfs:domain	ship:OutOfServiceVessel

Exercise 6

In the table below we can see that the ships have commanders. How can we assert that the commander of a ship is a captain? And that John and Alexander are therefore two captains?

Name	Maiden Voyage	Next Departure	Commander
MV 16	May 23, 2001	November 29, 2013	John
MV 22	June 8, 1970		Alexander

Table: Ships

Solution

ship:hasCommander	rdfs:range	ship:Captain
ship:John	rdf:type	ship:Captain
ship:Alexander	rdf:type	ship:Captain

References

- RDF Primer (W3C): http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/
- Resource Description Framework (RDF): Concepts and Abstract Syntax (W3C): http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
- RDF Schema (W3C): http://www.w3.org/TR/rdf-schema/
- G. Antoniou & F. van Harmelen (2004). A Semantic Web Primer (Cooperative Information Systems). MIT Press, Cambridge MA, USA.
- F. Giunchiglia, F. Farazi, L. Tanca, and R. D. Virgilio. The semantic web languages. In Semantic Web Information management, a model based perspective. Roberto de Virgilio, Fausto Giunchiglia, Letizia Tanca (Eds.), Springer, 2009.
- D. Allemang and J. Hendler. Semantic web for the working ontologist: modeling in RDF, RDFS and OWL. Morgan Kaufmann Elsevier, Amsterdam, NL, 2008.